MAP inference in dynamic hybrid Bayesian networks

Ramos-López, Darío ; Masegosa, Andres; Martinez, Ana M.; Salmerón, Antonio; Nielsen, Thomas Dyhre; Langseth, Helge; Madsen, Anders Læsø

Published in:
Progress in Artificial Intelligence

DOI (link to publication from Publisher):
10.1007/s13748-017-0115-7

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

Prog Artif Intell (2017) 6:133–144
DOI 10.1007/s13748-017-0115-7

REGULAR PAPER

MAP inference in dynamic hybrid Bayesian networks

Darío Ramos-López[1] · **Andrés R. Masegosa**[2] · **Ana M. Martínez**[3] · **Antonio Salmerón**[1] · **Thomas D. Nielsen**[3] · **Helge Langseth**[2] · **Anders L. Madsen**[3,4]

Received: 12 December 2016 / Accepted: 13 January 2017 / Published online: 27 January 2017

Abstract In this paper, we study the maximum a posteriori (MAP) problem in dynamic hybrid Bayesian networks. We are interested in finding the sequence of values of a class variable that maximizes the posterior probability given evidence. We propose an approximate solution based on transforming the MAP problem into a simpler belief update problem. The proposed solution constructs a set of auxiliary networks by grouping consecutive instantiations of the variable of interest, thus capturing some of the potential temporal dependences between these variables while ignoring others. Belief update is carried out independently in the auxiliary models, after which the results are combined, producing a configuration of values for the class variable along the entire time sequence. Experiments have been carried out to analyze the behavior of the approach. The algorithm has been implemented using Java 8 streams, and its scalability has been evaluated.

Keywords MAP inference · Hybrid Bayesian networks · Temporal models

✉ Antonio Salmerón
antonio.salmeron@ual.es

Darío Ramos-López
dramoslopez@ual.es

Andrés R. Masegosa
andresrm@idi.ntnu.no

Ana M. Martínez
ana@cs.aau.dk

Thomas D. Nielsen
tdn@cs.aau.dk

Helge Langseth
helgel@idi.ntnu.no

Anders L. Madsen
anders@hugin.com

[1] Department of Mathematics, University of Almería, Almería, Spain

[2] Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway

[3] Department of Computer Science, Aalborg University, Aalborg, Denmark

[4] Hugin Expert A/S, Aalborg, Denmark

1 Introduction

Data acquisition is nowadays ubiquitous in any technological environment, and large amounts of data are being produced, often to an extent where it becomes a major challenge to make use of it. In many contexts, uncertainty is inherent to the data for different reasons (for instance, measurement noise, technical limitations, incomplete information). Thus, suitable modeling and inference techniques are essential to make an adequate interpretation of the data. Probabilistic graphical models (PGMs) are known to be a well-founded and principled tool for performing inference and belief updating in complex domains endowed with uncertainty. In this work, we focus on Bayesian networks (BNs) [15], which constitute a particular type of PGMs. Specifically we will consider dynamic hybrid BNs (DBNs), which represent situations with a temporal dimension, and which allow discrete and continuous variables to coexist. We restrict our attention to conditional linear Gaussian (CLG) models [10,11].

In probabilistic reasoning, the MAP problem is of special relevance and difficulty [2,6,14]. It amounts to finding the most probable configuration (called MAP configuration) of some specific *variables of interest* (also called MAP variables), given observations of some *evidence variables*. This is sometimes also referred to as partial abductive inference [1].

An instantiation of this problem, which we will focus our discussion around, is *dynamic classification*. Here the variables of interest are the temporal replications of a discrete variable that at time t signifies the *classification of the sequence* at that time. Thus, the MAP configuration is used to obtain the most probable *development* of the class over a time sequence, given (partially) observed attributes. We are therefore interested in finding the values of the class variables that together maximize the (joint) posterior probability of the sequence given the evidence and do not focus on the correct classification at each single point in time (the two sequences are not necessarily the same). Typically, in dynamic classification, one is interested in detecting time periods of special relevance and detecting specific instants where changes in the class happen.

A direct approach to tackle this problem is to first "unroll" the DBN, that is, to convert it into a static model by including one copy of each variable for each time instance and then compute the MAP configuration in the unrolled model using existing algorithms for static BNs. Due to the computational complexity of the MAP calculations [14] and their inability to benefit from the regularities of the network structure of the unrolled dynamic model, this approach would, however, only be feasible for very short sequences. Another immediate idea is to use the well-known Viterbi algorithm [5]. However, this approach only works for calculating the *most probable explanation* (MPE), a special case of MAP where all the unobserved variables of interest.

Our proposal, which is an approximate solution based on transforming the dynamic MAP problem into simpler belief update problems, consists of two steps:

(i) Generate a collection of auxiliary models based on the original one. In each of these models, the temporal replications of the variable of interest are joined into compound nodes that capture some of the temporal dependencies among the temporal instantiations of the class variable. Then perform standard belief update separately in each of these models;
(ii) Combine the results to find a sequence of values for the variables of interest that approximates the MAP sequence.

By making operations on the *structure* of the original model, we thus approximately solve a MAP inference problem by performing a set of (simpler) probabilistic inference tasks. In this general formulation, any exact or approximate algorithm for belief update can be employed for the inference process in *i*).

The proposed algorithm has been implemented using the AMIDST Toolbox [13], and we exemplify its use by using variational message passing (VMP) [18] and importance sampling (IS) [7,8,17] for belief update. A set of experiments has been carried out to assess the suitability of the approach.

The rest of the paper is organized as follows: Sect. 2 establishes the necessary background and gives the precise formulation of the problem. Then, the proposed algorithm is detailed in Sect. 3. The performance of the algorithm is examined in Sect. 4, before the paper ends with conclusions in Sect. 5.

2 Preliminaries

In this section, we introduce the main mathematical concepts employed in the paper: those of (dynamic hybrid) BNs, MAP inference, and the dynamic MAP problem. A BN consists of a directed acyclic graph, where the nodes correspond to random variables and the arcs represent dependencies among them; for ease of presentation, we will use the terms node and variable interchangeably throughout this paper. Attached to each node in the graph, there is a conditional probability distribution for the corresponding variable given its parents in the network.

A BN with variables $\mathbf{X} = \{X_1, \ldots, X_N\}$ defines a joint probability distribution that factorizes as

$$p(\mathbf{X}) = \prod_{i=1}^{N} p(X_i | Pa(X_i)), \tag{1}$$

where $Pa(X_i)$ is the set of parents of X_i in the graph. A BN is called *hybrid* if it has both discrete and continuous variables. We will use lowercase letters to refer to values or configurations of values, so that x corresponds to a value of the variable X and boldface $\mathbf{x}$ refers to a configuration of the variables in $\mathbf{X}$.

A *CLG BN* is a hybrid BN, where the joint distribution is a CLG [11]. In a CLG BN, the conditional distribution of each discrete variable $X_D \in \mathbf{X}$ given its parents is a multinomial, while the conditional distribution of each continuous variable $Z \in \mathbf{X}$ with discrete parents $\mathbf{X}_D \subseteq \mathbf{X}$ and continuous parents $\mathbf{X}_C \subseteq \mathbf{X}$ is given as a normal density by

$$p(z|\mathbf{X}_D = \mathbf{x}_D, \mathbf{X}_C = \mathbf{x}_C) = \mathcal{N}(z; \alpha_{\mathbf{x}_D} + \boldsymbol{\beta}_{\mathbf{x}_D}^{\mathrm{T}} \mathbf{x}_C, \sigma_{\mathbf{x}_D}), \tag{2}$$

for all $\mathbf{x}_D \in \Omega_{\mathbf{X}_D}$ and $\mathbf{x}_C \in \Omega_{\mathbf{X}_C}$; $\alpha_{\mathbf{x}_D}$ and $\boldsymbol{\beta}_{\mathbf{x}_D}$ are the coefficients of a linear regression model of Z (there is one such set of coefficients for each configuration of the discrete parent variables).

Thus, the conditional mean of Z is a linear function of its continuous parents, while its standard deviation, σ_D, only depends on the discrete parents. Finally, it should be noted that a CLG BN does not allow continuous variables with discrete children.

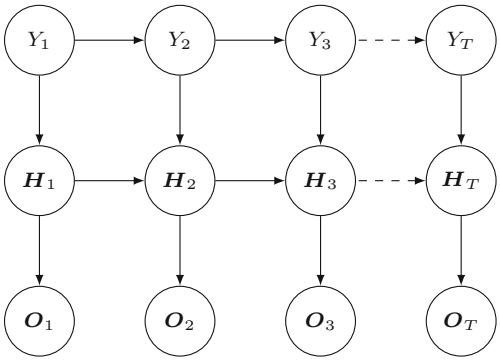

Fig. 1 Simplified dynamic BN

In this paper, we put special emphasis on *dynamic* CLG BNs, which explicitly model the evolution over time [3] by indexing the variables $\mathbf{X}_t$ with a discrete time stamp $t \in \{1, \ldots, T\}$; as a shorthand notation, we shall use $\mathbf{X}_{1:T} = (\mathbf{X}_1, \ldots, \mathbf{X}_T)$. Specifically, we will consider model structures as illustrated in Fig. 1, where each time step t is composed of a variable of interest, denoted Y_t, a set of hidden variables, $\boldsymbol{H}_t$, and a set of observable variables $\boldsymbol{O}_t$.

Note that the model adheres to the Markov assumption: At any time t, the variables prior to t, i.e., $\{Y_{1:t-1}, \boldsymbol{O}_{1:t-1}, \boldsymbol{H}_{1:t-1}\}$ are conditionally independent of the future variables $\{Y_{t+1}, \boldsymbol{O}_{t+1}, \boldsymbol{H}_{t+1}\}$ given the variables at time t, $\{Y_t, \boldsymbol{O}_t, \boldsymbol{H}_t\}$. By also assuming that the model is stationary (i.e., $p(X_t|Pa(X_t)) = p(X_s|Pa(X_s))$, for all time steps s and t), the dynamic model can be compactly specified through a prior model of the initial time step and a transition model for two consecutive time steps, i.e., a model of the distributions $p(\mathbf{X}_0)$ and $p(\mathbf{X}_t|\mathbf{X}_{t-1})$. Notice also that while Y_T is a single variable, both $\boldsymbol{O}_t$ and $\boldsymbol{H}_t$ are vectors. These vectors have an internal dependence structure that is not explicitly depicted in Fig. 1 and will not be utilized in the following except in the underlying belief update algorithms.

2.1 The MAP inference problem

Generally, for the MAP inference problem we look for a configuration of a particular subset of variables $\mathbf{Y}$ having maximum posterior probability given the observation $\mathbf{x}_E$ of another set of variables $\mathbf{X}_E$:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}\in\Omega_\mathbf{Y}} p(\mathbf{y}|\mathbf{x}_E) = \arg\max_{\mathbf{y}\in\Omega_\mathbf{y}} p(\mathbf{y}, \mathbf{x}_E).$$

If $\mathbf{X}_E \cup \mathbf{Y} \subset \mathbf{X}$, the variables in the set $\mathbf{X} \setminus (\mathbf{X}_E \cup \mathbf{Y})$ should be marginalized out before doing the maximization:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}\in\Omega_\mathbf{Y}} \sum_{\mathbf{x}_D\in\Omega_{\mathbf{X}_D\setminus(\mathbf{Y}\cup\mathbf{X}_E)}} \int_{\mathbf{x}_C\in\Omega_{\mathbf{X}_C\setminus(\mathbf{Y}\cup\mathbf{X}_E)}} p(\mathbf{y}, \mathbf{x}_D, \mathbf{x}_C, \mathbf{x}_E)\mathrm{d}\mathbf{x}_C.$$

In the special case where $\mathbf{X} = \mathbf{X}_E \cup \mathbf{Y}$, the inference problem is also referred to as finding the MPE.

When dealing with DBNs of the form in Fig. 1, the variables of interest are $\mathbf{Y} = (Y_1, \ldots, Y_T)$ and the MAP problem amounts to finding

$$\mathbf{y}^*_{1:T} = \arg\max_{\mathbf{y}_{1:T}\in\Omega_{\mathbf{Y}_{1:T}}} p(\mathbf{y}_{1:T}, \mathbf{o}_{1:T}),$$

which involves summing/integrating out the unobserved variables $\boldsymbol{H}_{1:T}$.

Finding an exact solution to the general MAP problem has been shown to be $\mathrm{NP}^{\mathrm{PP}}$ complete [14]. Furthermore, if N equals the total number of variables in the unrolled model (thus, $N = \kappa T$, where κ is the number of variables in $\mathbf{X}_t$), approximating MAP with an error-bound growing slower than exponentially in N is NP-hard [14,16].

General-purpose algorithms for MAP can therefore not be employed in our setting when T (and therefore N) can take on very large values, and we will rather pursue a tailor-made approach, which utilizes the particular structure of the unrolled BN. The two important features that we will take advantage of are i) the Markov assumption inherent in the model and ii) that the variables of interest are $Y_{1:T}$, with $Pa(Y_t) = \{Y_{t-1}\}$, $t = 2, \ldots, T$. Nevertheless, the problem is still computationally difficult to solve, as the model in Fig. 1 reveals that all the variables $\{Y_1, \ldots, Y_T\}$ are dependent conditional on $\mathbf{O}_{1:T}$. Hence, the posterior distribution $p(\mathbf{y}_{1:T}|\mathbf{o}_{1:T})$ cannot be represented in factorized form.

3 A MAP algorithm for DBNs

In what follows, we present a framework for approximately solving the MAP problem for DBNs by exploiting the temporal structure of the model. We seek to reduce the MAP inference problem to the problem of performing belief update in a collection of secondary compound model structures derived from the original DBN. The framework is independent of the particular belief update procedure being used, and the updating procedure can thus be selected based on the particularities of the domain being modeled.

The basic idea is to find a MAP configuration through a two-step process. First, we estimate posterior distributions over subsets of MAP variables (i.e., subsets of $\mathbf{Y}_{1:T}$); by looking at subsets of variables, we capture some of the dependencies of the posterior distribution. In the second step, we combine these distributions to obtain an approximation of the joint posterior distribution. Based on this compound distribution, we seek a MAP configuration. If the subsets used in the initial step are singletons, we have a fully factorized posterior distribution for which the MAP configuration corresponds to independently identifying the most probable state for each MAP variable. On the other hand, if the (single)

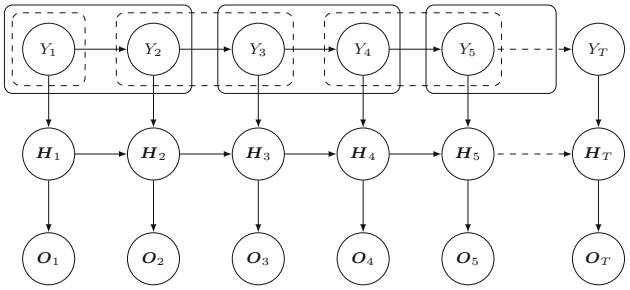

Fig. 2 Two possible partitions for pair-wise MAP variables merging are marked in *solid* and *dashed lines*

subset used initially consists of $\mathbf{Y}_{1:T}$, our algorithm is identical to exactly recovering the MAP sequence. Intermediate solutions will have most relevance in practice.

For ease of presentation, the below description of the method will be given relative to the simplified model structure shown in Fig. 1.

3.1 Step 1: Compound model specification

As outlined above, we transform the MAP problem into a collection of simpler inference problems over derived structures based on which an estimate of the MAP sequence is obtained. The aim of the derived structures is to capture a subset of the potential dependencies between the MAP variables. For this purpose, we consider grouping consecutive MAP variables into compound variables under the assumption that temporally close MAP variables are more strongly dependent than those further apart in the sequence.

If we only consider pair-wise dependencies between consecutive MAP variables, then this procedure will give rise to two different partitionings as illustrated in Fig. 2; the partitionings are depicted with solid lines and dashed lines, respectively. These two groupings not only capture different dependency relations between consecutive MAP variables, they also represent the only possible pair-wise consecutive partitionings of the MAP variables. The compound model derived by merging nodes according to the partitioning defined by the solid lines is represented in Fig. 3 (top). Each compound node Z_i in this model is defined as the Cartesian product of its two constituent MAP variables Y_{2i} and Y_{2i+1}; for the dashed partitioning, the compound node Z_i would correspond to (Y_{2i-1}, Y_{2i}) and with Z_0 being a singleton (see Fig. 3 (bottom)). The same construction scheme can be applied when merging 3, 4, or in general D consecutive copies of the MAP variables, producing a corresponding number of auxiliary models.

In order to complete the specification of the model in Fig. 3 (top), we need to populate it with probability distributions. To simplify notation, let $Z_{j-1,j}$ denote the aggregated variable corresponding to $Y_{j-1} \times Y_j$ and let $\boldsymbol{H}_k$ be a child of $Z_{j-1,j}$ (thus $k = j-1$ or $k = j$). For these variables, the required

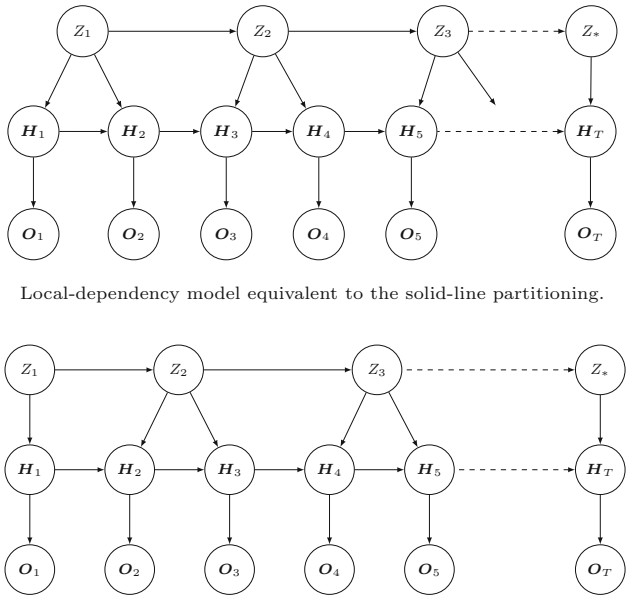

Fig. 3 Two possible pair-wise partitionings and combinations of consecutive MAP variables

distributions can readily be calculated based on the original model as

$$
\begin{aligned}
&P_i\big(Z_{j-1,j} = (y_{j-1}, y_j) | Z_{j-3,j-2} = (y_{j-3}, y_{j-2})\big) \\
&\quad = P(y_j | y_{j-1}) P(y_{j-1} | y_{j-2}), \\
&P_i\big(\boldsymbol{H}_k | Z_{j-1,j} = (y_{j-1}, y_j)\big) \\
&\quad = \begin{cases} P(\boldsymbol{H}_k | Y_j = y_{j-1}) & \text{if } k = j-1, \\ P(\boldsymbol{H}_k | Y_j = y_j) & \text{if } k = j. \end{cases}
\end{aligned}
$$

The remaining distributions in the aggregated model are directly copied from the original DBN. A similar specification of the probability distributions for the compound model in Fig. 3 (bottom) follows the same pattern as above but with shifted indexes.

Using the aggregated models, we can perform standard belief update using, for instance, mean-field variational inference or importance sampling, and still capture local dependencies between the MAP variables in the original model. Based on these posterior distributions, we approximate the joint posterior distribution of the MAP variables based on which the inference scheme can efficiently be performed.

3.2 Step 2: Model combination and MAP inference

Let $Z^{(i)}_{1:|\mathcal{P}_i|}$ be the $|\mathcal{P}_i|$ compound variables for the ith partitioning $\mathcal{P}_i$. Now, we denote with $P_i\left(Z^{(i)}_{1:|P_i|} | \boldsymbol{o}_{1:T}\right)$ the posterior distribution of the MAP variables $\boldsymbol{Y}_{1:T}$ in $\mathcal{P}_i$; the distribution is defined over the induced compound variables.

The size of this probability table is $O\left(2^T\right)$ assuming that Y_t is binary. Thus, even if the posterior can be calculated in theory, it is infeasible in practice. Instead, we will resort to using the approximate marginal posterior

$$\tilde{P}_i(Y_1, \ldots, Y_T|\mathbf{o}_{1:T}) = \prod_{j=1}^{|\mathcal{P}_i|} P_i(Z_j^{(i)}|\mathbf{o}_{1:T}). \tag{3}$$

The posterior distributions obtained for the different partitionings encode distinct dependencies among the MAP variables. For instance, for the two partitionings in Fig. 2 we have that $Y_2 \perp\!\!\!\perp_{P_2} Y_3|\mathbf{o}_{1:T}$ and $Y_2 \not\perp\!\!\!\perp_{P_1} Y_3|\mathbf{o}_{1:T}$, where $\perp\!\!\!\perp_{P_2}$ and $\perp\!\!\!\perp_{P_1}$ are the independence relations captured by the posterior distributions defined according to Eq. (3) for the solid and dashed partitioning, respectively. In what follows, we will consider a partitioning of the MAP variable to be synonymous with the compound model it induces and the two terms will therefore be used interchangeably.

Individually, each compound model captures only a subset of the posterior dependencies over $\mathbf{Y}_{1:T}$, but collectively they define a richer dependency model as compared to any of the individual compound models. Thus, we seek to compute a MAP configuration over a *joint* posterior $P^*(Y_1, \ldots, Y_T|\boldsymbol{o}_{1:T})$ which combines the local posteriors obtained for the different partitions. Assuming that we have D partitions, we choose P^* as the *centroid* distribution using the KL divergence as a distance measure,

$$P^* = \arg\min_P \sum_{k=1}^{D} KL(P_i, P).$$

Following the results presented in [12, Theorem 3.2], there is a closed-form solution for the above problem. In the case of multinomial variables, the solution simplifies to

$$P^*(Y_1, \ldots, Y_T|\boldsymbol{o}_{1:T}) = \frac{1}{D}\sum_{i=1}^{D} \tilde{P}_i(Y_1, \ldots, Y_T|\boldsymbol{o}_{1:T}). \tag{4}$$

Based on this centroid distribution P^*, we can now find an approximation to the original MAP problem by finding a configuration maximizing $P^*(Y_1, \ldots, Y_T|\boldsymbol{o}_{1:T})$. From Eqs. (3) and (4), we have that, for the partitioning in Fig. 2, the centroid distribution defines a first-order Markov model over $Y_1, \ldots, Y_T$, where the conditional distributions $P^*(Y_i|Y_{i-1})$ are given by[1]

$$\begin{aligned} P^*(Y_i|Y_{i-1}) &\propto \sum_{\mathbf{Y}_{1:T}\setminus\{Y_i, Y_{i-1}\}} P^*(Y_1, \ldots, Y_T) \\ &= \frac{1}{2}(P_1(Y_{i-1}, Y_i) + \sum_{Y_{i-2}} P_2(Y_{i-2}, Y_{i-1}) \sum_{Y_{i+1}} P_2(Y_i, Y_{i+1})), \end{aligned} \tag{5}$$

assuming that partitioning $\mathcal{P}_1$ defines the subset $\{Y_{i-1}, Y_i\}$; otherwise, the indexes should be reversed.

An approximate MAP configuration can now be found by simply applying the standard Viterbi algorithm [5] based on the Markov model defined according to Eq. (5).

The procedure described above can be generalized to higher-order partitions, thereby potentially producing an even richer dependency model. That is, rather than working with pair-wise relations, one could instead consider grouping an arbitrary number D of consecutive MAP variables. This would produce D different compound models, which, in turn, would induce a $(D-1)$-order Markov chain over the original replications of the MAP variable. Again, an approximate MAP sequence can be obtained using the Viterbi algorithm with this higher-order Markov chain.

Algorithm 1 summarizes our proposal for MAP inference over DBNs.

Dynamic_MAP(B,$\mathbf{o}_{1:T}$,D)
Input: A dynamic BN B with variable of interest $Y_{1:T}$, a sequence of observations $\mathbf{o}_{1:T}$ and the number D of compound models to use.
Result: An approximate MAP estimate.
1 Unroll B over time steps $1, \ldots, T$.
2 **for** $i \leftarrow 1$ **to** D **do**
3 Compute a compound model M_i as described in Sect. 3.1.
4 Compute the posterior distribution
$\tilde{P}_i(Y_1, \ldots, Y_T|\mathbf{o}_{1:T})$
as in Equation (3) using compound model M_i.
5 **end**
6 Compute
$$P^*(Y_1, \ldots, Y_T|\boldsymbol{o}_{1:T}) = \frac{1}{D}\sum_{i=1}^{D} \tilde{P}_i(Y_1, \ldots, Y_T|\boldsymbol{o}_{1:T}).$$
7 Obtain a MAP configuration $\mathbf{y}^*_{1:T}$ from $P^*(Y_1, \ldots, Y_T|\boldsymbol{o}_{1:T})$ using the Viterbi algorithm.
8 **return** $\mathbf{y}^*_{1:T}$

Algorithm 1: The dynamic MAP algorithm.

4 Experimental evaluation

In order to evaluate our proposal, we have conducted a series of experiments in two example models; each one is specified by a DBN, whose structure is in accordance with the general model structure introduced in Fig. 1.

[1] For ease of presentation, we drop the explicit conditioning on $\boldsymbol{o}_{1:T}$ in the equations below.

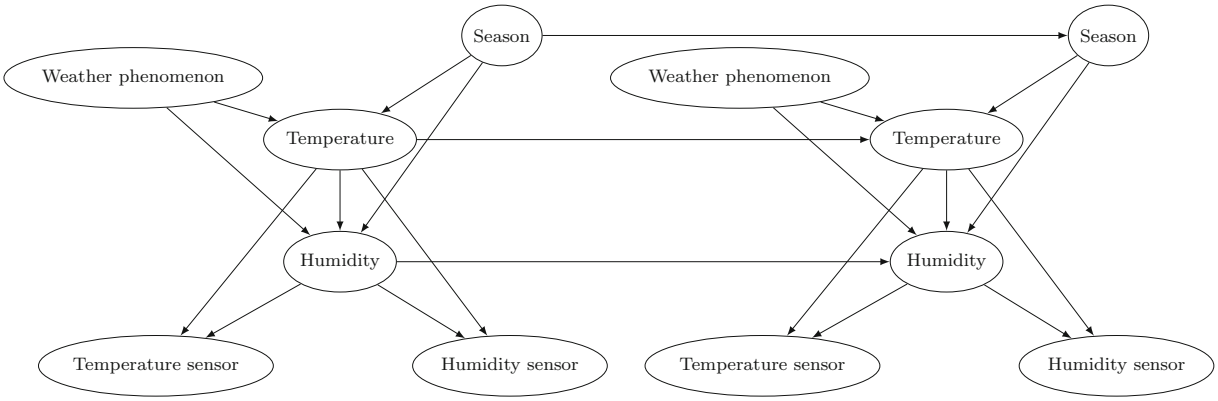

Fig. 4 Sample two-time slice DBN for detecting season change

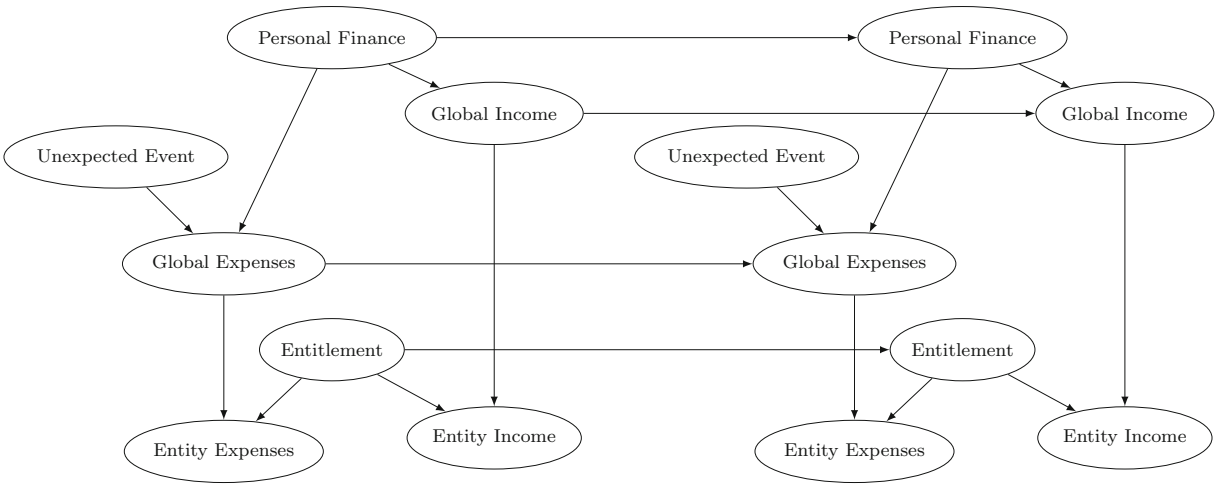

Fig. 5 Sample two-time slice DBN for detecting change in personal finances

The first model detects seasonal changes based on measurements of temperature and humidity. It includes the season (the class variable), the presence of a special weather phenomenon (such as the sudden appearance of a heat/cold front), the true temperature and humidity values as hidden variables, and noisy sensor readings of the temperature and humidity as the observable variables. In accordance with the general model structure, only the hidden variables and the class explicitly model dynamics; notice however that the variable related to weather phenomenons is not temporally connected over time. A two-time slice representation of the model is illustrated in Fig. 4.

The second model monitors the financial status of a bank client in order to identify periods of time where the client is having financial difficulties. One could envision that a bank would be interested, for instance, in gathering additional information about the situation in order to decrease the risk of client defaulting, or in offering special products that could help the client recover from the situation (e.g., debt refinancing). Figure 5 provides an illustration of the model structure. The DBN has the financial situation of the client as the class variable; the overall income and expense level (including all other financial obligations or assets), the appearance of an unexpected event that could affect the personal situation, and the engagement level of the client with respect to the bank as unobserved variables; and her/his expenses and incomes at this specific bank as observable variables.

Both models involve discrete and continuous (Gaussian) variables. The hybrid nature of the models together with the fact that some of the variables are unobserved precludes exact marginal inference in general. For instance, marginalizing out variable weather phenomenon from the season change model yields a mixture of two CLG models. In general, the number of mixture terms grows exponentially in the number of hidden (unobserved) discrete variables.

For the analysis of the proposed framework, we have considered two instantiations that differ wrt. the underlying belief update algorithm, VMP or IS, when computing the posterior distribution in Step 4 of Algorithm 1. The implementation has been designed taking scalability into account from two perspectives. On the one hand, the implementation is scalable in terms of the number of compound models

used. It can be seen how Algorithm 1 follows a Map/Reduce scheme where the Map phase corresponds to Steps 3 and 4 and the Reduce phase corresponds to Step 6. Additionally, the implementation takes advantage of the scalability of the implementation of the IS method provided by the AMIDST Toolbox [13,17], on top of which our implementation is built.

We compare the results with two baseline models: Firstly, for short sequences (up to approximately 10 time steps), the exact solution of the dynamic MAP problem can be obtained using the HUGIN system [9]. HUGIN does exact inference in the unrolled network, and the approximate MAP sequence provided by our scheme can therefore be compared to the correct MAP sequence whenever HUGIN is able to provide a result. Secondly, we utilize a simple baseline akin to greedy search: We start by doing standard belief update in the model with evidence $\mathbf{x}_E$, then locate the MAP variable with smallest entropy, and select its most probable value. This selection is considered new evidence, so we do a new belief update, and keep selecting configurations for the variables of interest until all of them are instantiated. The selected configuration is the approximate MAP sequence. This approach is called "iterative," and it can be run with any of the inference methods (HUGIN, when available, IS, or VMP). This procedure is similar to the so-called *sequential initialization* in [14].

Several experiments have been performed to evaluate the algorithm along two axes: First, we consider the ability of the proposed method to recover the MAP sequence given the evidence; second, we analyze the computational complexity and scalability of the method. We do this by randomly generating problems from the two models as follows: The model is unrolled to contain T time steps, and *all* values of all variables are simulated using forward sampling. The sampled values for the *observable variables* are retained; these values define the evidence $\mathbf{x}_E$ which is provided to the algorithms. The procedure is repeated a number of times to provide robust results.

It should be mentioned here that evaluating the method's ability to recover the MAP sequence can be problematic. For short sequences, where the HUGIN system is able to calculate the correct MAP sequence and the likelihood of a particular sequence can be precisely estimated, we compute for each method the likelihood ratio with respect to the actual MAP sequence given by HUGIN. In addition, we compare the correct and proposed sequences (both wrt. an exact match and the average precision for each model, using several metrics). One possibility is measuring the precision as 1 minus the normalized Hamming distance between the sequences, so that 1 is a perfect match. We will refer to this metric as the Hamming-based accuracy.

Instead of comparing the sequences element-wise, one can analogously compare them using moving windows, i.e., comparing subsequences of length $s \geq 1$, leading to what we will call s-tuple Hamming-based accuracy. When s is equal to the sequence length T, this boils down to the 0–1 accuracy (1 if the two sequences fully match, 0 otherwise). For longer sequences, however, we do not have access to the correct MAP sequence. Furthermore, we also do not have access to the probability $p(\mathbf{x}_I^*|\mathbf{x}_E)$ of a given sequence $\mathbf{x}_I^*$, as calculating this probability in general involves working with mixture models where the number of components grows exponentially in T (due to the hidden variables in the models). Therefore, we use the *sampled* sequences restricted to the MAP variables as they were the MAP sequences and calculate the previously introduced accuracy metrics based on the Hamming distance between those series and the recovered approximate MAP sequences. It is important to notice that in the case of short sequences, the most meaningful metric is the likelihood ratio, where the density of each approximated MAP sequence is compared to that of the exact one. However, in the experiments corresponding to longer sequences, where the comparison is based on the sampled sequence, the precision values may be less meaningful and should only be considered for relative comparison between the different algorithms.

For the first experiment, we generated data as described above. Then, the dynamic MAP algorithm was run with different group configurations (grouping 2, 3, or 4 consecutive MAP variable replications), and the estimated MAP sequence was obtained for each of them. Also, the marginal MAP approach (with no grouping of MAP variable replications, called *ungrouped* in the tables and figures) and the entropy-based baseline (*iterative* in the tables) have been run for comparison.

For short sequences (7 or 9 time steps, depending on the model), 50 different repetitions of the experiment were carried out and the average evaluation metrics precision values (described above) for each model appear in Tables 1 and 2. Here, the sequences are compared with the originally sampled sequence with different metrics, and the correct MAP sequence given by HUGIN in terms of the likelihood ratio (see last row of each model). According to these results, the metrics that are more correlated with the likelihood ratio are the Hamming-based and the 3-tuple Hamming-based accuracies, as high values of them correspond also to high likelihood ratios. Thus, in cases where the correct MAP sequence cannot be found, these metrics (wrt. the sampled sequence) might provide an approximate manner of evaluating the precision. In addition, the 0–1 distance does not match well with our purpose of detecting time periods of special relevance (in the examples, weather phenomena or financial difficulties), since the aim is not recovering the whole exact sequence but detecting some specific instants where changes happen.

The results in Tables 1 and 2 show how IS is more accurate than VMP for the financial situation model, while VMP is the most accurate for the season change model. We conjecture that this difference is caused by the presence of extreme

Table 1 Analysis of the precision of the approximate dynamic MAP methods in short sequences, compared to the originally sampled sequence

Version	Season change model (9 time steps)		
	IS	VMP	HUGIN
Hamming-based accuracy			
Iterative	0.715	0.777	0.793
Ungrouped	0.680	0.780	0.793(*)
2-Grouped	0.713	0.787	
3-Grouped	0.753	0.782	
4-Grouped	0.731	0.784	
3-Tuples Hamming-based accuracy			
Iterative	0.491	0.577	0.597
Ungrouped	0.400	0.574	0.597(*)
2-Grouped	0.440	0.583	
3-Grouped	0.528	0.580	
4-Grouped	0.494	0.583	
0–1 distance accuracy			
Iterative	0.26	0.24	0.30
Ungrouped	0.10	0.24	0.30(*)
2-Grouped	0.14	0.24	
3-Grouped	0.22	0.24	
4-Grouped	0.18	0.24	
Likelihood ratio (wrt. the correct MAP seq.)			
Iterative	0.752	0.933	1.000
Ungrouped	0.351	0.948	1.000(*)
2-Grouped	0.459	0.948	
3-Grouped	0.612	0.948	
4-Grouped	0.659	0.948	

Season change model
Numbers marked with (*) correspond to the correct MAP sequence given by HUGIN, i.e., equivalent to *9-Grouped*, since $T = 9$

Table 2 Analysis of the precision of the approximate dynamic MAP methods in short sequences, compared to the originally sampled sequence

Version	Financial situation model (7 time steps)		
	IS	VMP	HUGIN
Hamming-based accuracy			
Iterative	0.811	0.651	0.866
Ungrouped	0.849	0.660	0.869(*)
2-Grouped	0.869	0.665	
3-Grouped	0.860	0.671	
4-Grouped	0.857	0.671	
3-Tuples Hamming-based accuracy			
Iterative	0.652	0.52	0.772
Ungrouped	0.708	0.532	0.768(*)
2-Grouped	0.764	0.544	
3-Grouped	0.740	0.548	
4-Grouped	0.740	0.548	
0–1 distance accuracy			
Iterative	0.40	0.24	0.58
Ungrouped	0.48	0.24	0.58(*)
2-Grouped	0.60	0.26	
3-Grouped	0.58	0.26	
4-Grouped	0.54	0.26	
Likelihood ratio (wrt. the correct MAP seq.)			
Iterative	0.697	0.569	0.998
Ungrouped	0.811	0.569	1.000(*)
2-Grouped	0.878	0.586	
3-Grouped	0.909	0.586	
4-Grouped	0.952	0.586	

Financial situation model
Numbers marked with (*) correspond to the correct MAP sequence given by HUGIN, i.e., equivalent to *7-Grouped*, since $T = 7$

probabilities in the season change model, which increases the variability of the sampling process carried out by IS making it more prone to error. A detailed discussion on the sensitivity of IS to extreme probabilities can be found in [4].

For longer sequences (50, 100 and 200 time steps), each experiment was replicated 100 times and the average metrics are presented in Table 3. We could not obtain exact results using HUGIN for these sequences. This is due to the complexity of the resulting networks after unrolling. Recall that the number of variables in the unrolled network is equal to the number of variables in a time slice times the number of time steps. It means that a sequence of length 7 on the season change model results in an unrolled network with 42 variables, while a sequence of 200 time steps contains 1,200 variables. In order to obtain comparable results in terms of accuracy, the number of samples used by IS was increased according to model complexity and sequence length: For the season model with 9 time steps and the financial situation model with 7 time steps, 10,000 samples were used. For longer sequences, the same number of samples was used for both models: For 50 time steps, 20,000 samples; for 100 time steps, 50,000 samples; and for 200 time steps, 100,000 samples were simulated.

Table 4 reports run times of each method for short sequences, as well as HUGIN's [9] run time, showing that our approach is several orders of magnitude quicker. The experiments were run on a dual-processor AMD Opteron 2.8GHz server with 32 cores and 64GB of RAM, running Linux Ubuntu 14.04.1 LTS.

For long sequences (50, 100 and 200 time steps), we compare the run time of VMP and IS for the ungrouped and the 4-grouped versions on the season change model (similar results are obtained for the financial model). Results are reported in Fig. 6, which also plots the run time of IS when

Table 3 Mean precision of the approximate dynamic MAP sequence for longer sequences, compared to the originally sampled sequence

Version	Season change model						Financial situation model					
	Length 50		Length 100		Length 200		Length 50		Length 100		Length 200	
	IS	VMP	IS	VMP	IS	VMP	IS	VMP	IS	VMP	IS	VMP
Hamming-based accuracy												
Iterative	0.760	0.614	0.713	0.660	–	–	0.785	0.745	0.844	0.795	–	–
Ungrouped	0.797	0.617	0.889	0.679	0.862	0.680	0.885	0.757	0.951	0.816	0.905	0.795
2-Grouped	0.826	0.700	0.914	0.747	0.868	0.742	0.893	0.818	0.943	0.839	0.907	0.824
3-Grouped	0.845	0.735	0.909	0.764	0.873	0.757	0.898	0.835	0.946	0.855	0.912	0.843
4-Grouped	0.847	0.792	0.905	0.791	0.878	0.787	0.898	0.859	0.943	0.896	0.912	0.877
3-Tuples Hamming-based accuracy												
Iterative	0.716	0.505	0.699	0.575	–	–	0.752	0.713	0.832	0.785	–	–
Ungrouped	0.735	0.509	0.867	0.594	0.842	0.600	0.838	0.724	0.942	0.807	0.891	0.784
2-Grouped	0.766	0.620	0.893	0.693	0.847	0.689	0.855	0.786	0.935	0.829	0.892	0.814
3-Grouped	0.800	0.668	0.892	0.715	0.857	0.708	0.868	0.805	0.939	0.845	0.901	0.831
4-Grouped	0.806	0.730	0.890	0.747	0.864	0.744	0.868	0.829	0.935	0.886	0.903	0.867
0–1 distance accuracy												
Iterative	0.11	0.10	0.24	0.07	–	–	0.27	0.20	0.4	0.16	–	–
Ungrouped	0.11	0.10	0.21	0.07	0.07	0.0	0.34	0.22	0.63	0.18	0.25	0.05
2-Grouped	0.17	0.10	0.23	0.07	0.11	0.0	0.38	0.25	0.61	0.18	0.23	0.08
3-Grouped	0.18	0.13	0.24	0.08	0.11	0.0	0.45	0.28	0.64	0.22	0.29	0.09
4-Grouped	0.22	0.14	0.26	0.08	0.10	0.0	0.43	0.30	0.67	0.24	0.31	0.10

parallelizing the random sampling generation step (the most time consuming part), using the multi-core implementation provided by the AMIDST toolbox [13]. We show results using 1, 2, 4, 8, 16 or 32 cores. Figure 6 also details the speed-up factor (understood as the run time using 1 core divided by the run time using more cores) for IS. All methods were run 50 times with each number of cores, and the run time was measured covering the whole procedure: merging of variables, belief update on each partition, and final computation of the MAP sequence.

We can see that VMP is much quicker than IS on long sequences when using a single CPU. However, IS is able to exploit multi-core CPUs and reduces its run time dramatically, specially for the longest sequences (it achieves a speed-up factor of 25 for 32 cores on 200-size sequences). As a result, when using 32 cores, the computation time of IS is quite close to the one obtained with VMP. Notice that we have used a sequential implementation of VMP, while the implementation of IS provided by the AMIDST Toolbox is able to take advantage of the existence of multiple computing units by distributing the sample generation process among them [17]. In the experiments reported here, we have not analyzed the scalability in terms of distributing the calculations with the different compound models, as its number is typically low.

Table 4 Dynamic MAP run times on short sequences compared to HUGIN

Version	Average execution time (s)		
	IS	VMP	HUGIN
Season change model (sequences of length 9)			
Ungrouped	0.0797	0.0332	171.52
2-Grouped	0.1652	0.0606	
3-Grouped	0.1748	0.1136	
4-Grouped	0.2352	0.2433	
Financial situation model (sequences of length 7)			
Ungrouped	0.0682	0.0230	28.17
2-Grouped	0.1396	0.0490	
3-Grouped	0.1673	0.0697	
4-Grouped	0.2095	0.1027	

From the results obtained for short sequences (precisions in Tables 1 and 2, run times in Table 4), we conclude that an approximation to the dynamic MAP sequence can be obtained using the proposed algorithm. Furthermore, the results show that increasing the degree of the compound variables always increases the run time and typically the precision goes up as well. These results are to be expected, since the model with a single compound variable consisting of $D = T$ variables recaptures the full MAP problem. On

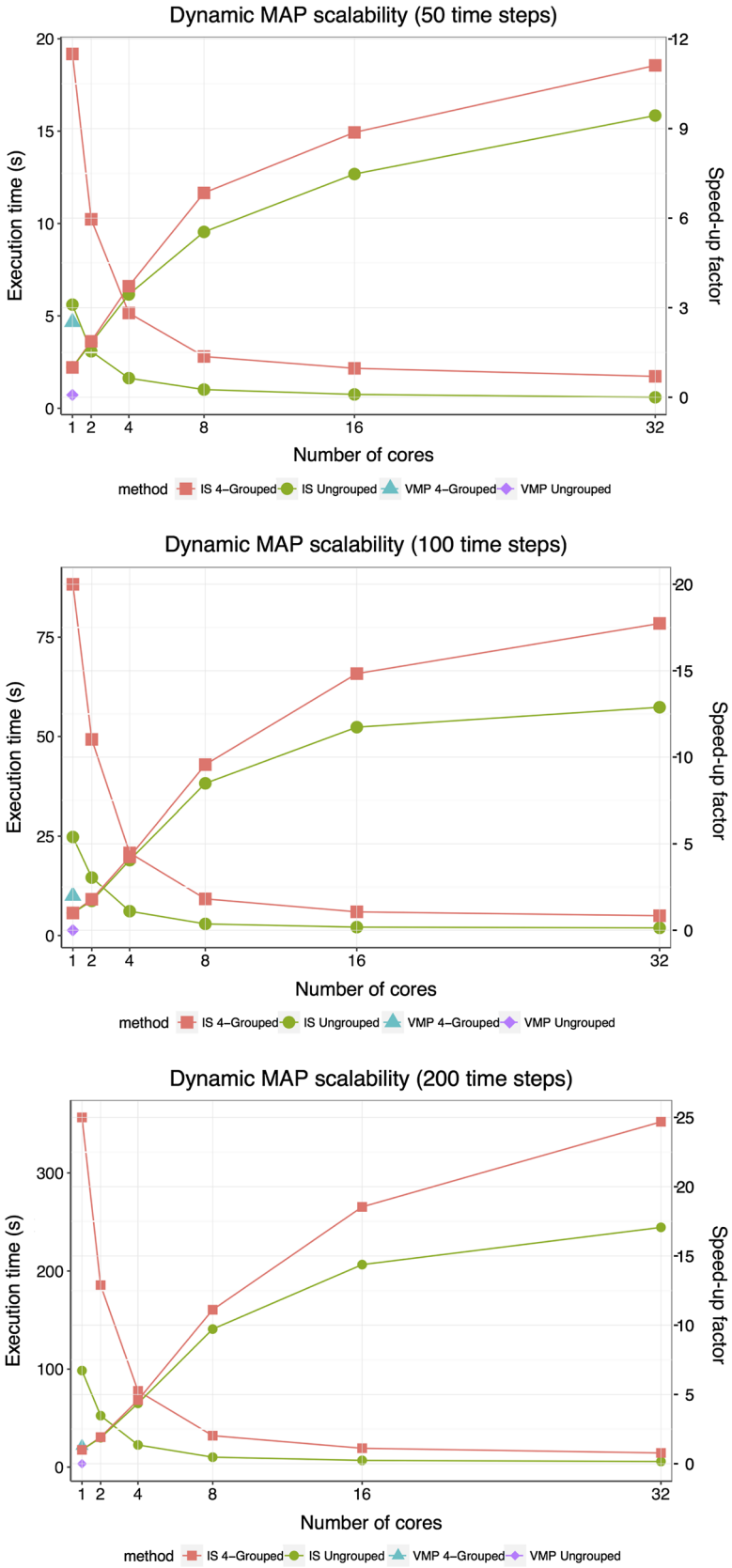

Fig. 6 Mean run times to obtain estimated MAP sequences of length 50, 100 and 200, versus the number of cores employed

the other hand, choosing $D = 1$ corresponds to marginal MAP. The corresponding results are denoted "Ungrouped" in the tables. Using degree up to $D = 4$, we find that the calculations are much faster than HUGIN's algorithm.

These results are consistent with those for longer sequences, see Table 3. However, the run time and space complexity of HUGIN's algorithm prevent us from comparing these results to the exact MAP sequences, and the results are therefore harder to interpret. In particular, Table 3 does not show a significant increase in the accuracy for IS when the variables are compounded, while the improvements are easily recognizable for VMP.

5 Conclusion

We have proposed a technique for solving the MAP problem in hybrid DBNs. As this problem is NP^{PP} complete in general, we resort to an approximate technique. The main idea is to translate the original MAP inference problem into the simpler problem of performing belief update in auxiliary networks, where the MAP variables are merged into compound nodes. The result of these belief updates is combined using their centroid distribution, and the Viterbi algorithm is then run to generate the final result.

We have first tested our approach on short sequences to be able to compare the obtained results to those generated by the state-of-the-art HUGIN software. This gave promising results. Our proposal is orders of magnitude faster than HUGIN and is mostly able to find similar sequences in terms of distance and likelihood ratio. The results also indicated that using more expressive compound variables resulted in improved accuracy and that the proposal improves the naïve marginal MAP baseline as well as the entropy-based baseline.

For larger models, the time and memory requirements of HUGIN made the comparison between our approximate scheme and the tool impractical. Additionally, calculating the exact log-probability of the generated sequences was infeasible. Therefore, we compared our models using metrics based on the Hamming distance between the generated sequence and the sampled one as a proxy. When the correct MAP sequence can be calculated, these metrics correlate well with the respective likelihood ratio. The results are still promising, but the lack of a well-defined gold-standard solution leaves them difficult to interpret.

We considered the parallelization of the sample generation step of IS. By doing that, we showed that IS and VMP have similar computational time complexity in practice. Overall, we found that our proposal provides useful sequences in short time.

Acknowledgements This work was partly carried out as part of the AMIDST project. AMIDST has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under Grant Agreement No 619209. This research has been partly funded by the Spanish Ministry of Economy and Competitiveness, through Project TIN2013-46638-C3-1-P and by ERDF funds.

References

1. de Campos, C.P.: New complexity results for MAP in Bayesian networks. In: Walsh, T. (ed.) Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, pp. 2100–2106 (2011)
2. de Campos, L.M., Gámez, J.A., Moral, S.: Partial abductive inference in Bayesian belief networks by simulated annealing. Int. J. Approx. Reason. **27**, 263–283 (2001)
3. Dean, T., Kanazawa, K.: A model for reasoning about persistence and causation. Comput. Intell. **5**(2), 142–150 (1989)
4. Fernández, A., Rumí, R., Salmerón, A.: Answering queries in hybrid Bayesian networks using importance sampling. Decis. Support Syst. **53**, 580–590 (2012)
5. Forney, G.: The Viterbi algorithm. Proc. IEEE **61**, 268–278 (1973)
6. Fu, Q., Wang, H., Banerjee, A.: Bethe-ADMM for tree decomposition based parallel MAP inference. In: Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI2013), pp. 222–231 (2013)
7. Fung, R., Chang, K.C.: Weighting and integrating evidence for stochastic simulation in Bayesian networks. In: Henrion, M., Shachter, R., Kanal, L., Lemmer, J. (eds.) Uncertainty in Artificial Intelligence, vol. 5, pp. 209–220. North-Holland, Amsterdam (1990)
8. Hammersley, J., Handscomb, D.: Monte Carlo Methods. Chapman & Hall, Boca Raton (1964)
9. Jensen, F.: Hugin API-reference manual, version 8.3. Hugin Expert A/S, Aalborg (2016)
10. Lauritzen, S.L., Jensen, F.: Stable local computation with conditional Gaussian distributions. Stat. Comput. **11**(2), 191–203 (2001)
11. Lauritzen, S.L., Wermuth, N.: Graphical models for associations between variables, some of which are qualitative and some quantitative. Ann. Stat. **17**, 31–57 (1989)
12. Liu, Q., Ihler, A.T.: Distributed estimation, information loss and exponential families. In: Advances in Neural Information Processing Systems, pp. 1098–1106 (2014)
13. Masegosa, A.R., Martínez, A.M., Borchani, H., Ramos-López, D., Nielsen, T.D., Langseth, H., Salmerón, A., Madsen, A.L.: AMIDST: analysis of massive data streams. In: Proceedings of the 27th Benelux Conference on Artificial Intelligence (BNAIC 2015) (2015)
14. Park, J., Darwiche, A.: Complexity results and approximation strategies for MAP explanations. J. Artif. Intell. Res. **21**, 101–133 (2004)

15. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., San Mateo (1988)
16. Roth, D.: On the hardness of approximate reasoning. Artif. Intell. **82**, 273–302 (1996)
17. Salmerón, A., Ramos-López, D., Borchani, H., Masegosa, A.R., Fernández, A., Langseth, H., Madsen, A.L., Nielsen, T.D.: Parallel importance sampling in conditional linear Gaussian networks. CAEPIA'2015. Lect. Notes Artif. Intell. **9422**, 36–46 (2015)
18. Winn, J., Bishop, C.: Variational message passing. J. Mach. Learn. Res. **6**, 661–694 (2005)